A VIRTUAL REALITY APPLICATION FOR SOFTWARE VISUALIZATION

Haruo Amari*, Toshiki Nagumo*, Mikio Okada*, Michitaka Hirose**, & Takemochi Ishii***

*Computer & Communication Research Center, Tokyo Electric Power Co., Inc.
1-4-10, Irifune, Chuo-ku, Tokyo 104, Japan
**Faculty of Engineering, University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo 113, Japan
***Faculty of Environmental Information, Keio University
5322, Endo, Fujisawa 252, Japan

1. Introduction

Demands for electric power have progressively increased during recent years. Accordingly, power supply networks have tended to become increasingly large and complex. Under these circumstances, the automatic control of complex power networks has become an important issue for system designers and software developers in order to ensure stability and to maintain the reliability of power supply. Since a power network control system constitutes a large-scale computer network, intricate functions of software such as processing synchronization and data transmission among the constituent computers are required for improvement of system performance and efficiency. However, at the present time, although software development tools for a single processor have been provided in many industrial sectors, methods for the support of overall large-scale system design are still scarce. In general, conventional text-oriented tools are insufficient to design, construct, and maintain large and complicated systems.

As a means for solving this problem, an analytical method was designed based upon computer graphics visualization of static structural data and execution trace data of a software application's functional units[1]. Furthermore, prototyping tools for the development of software enabling the direct manipulation of graphical elements through user-machine interfaces based upon Virtual Reality technology were developed[2,3].

2. Visual Analysis and Visual Design

In general, software visualization, also known as visual programming, can be classified into two major categories: 1) Visual Environments which show program construction and execution for software design and understanding, and 2) Visual Languages which handle visual information and allow programming with visual representations[4]. The present study covers both categories. Two types of operations were established as methods for software development support through visualization: 1) Visual Analysis, which includes functions for displaying software program structure and execution results on computer network systems, and 2) Visual Design, which includes functions for creating and editing of visual representations of software and transmitting the updated information in the software structure to the target machines (see Figure 1).

To execute these functions, tools were created for two different computer systems; a graphics workstation employed for visualization, and a general-purpose workstation employed for preparation of data. The tools created for the general purpose workstation were referred to as "Analytical Tools", and those created for the graphics workstation were referred to as parts of a "Virtual Reality Environment." The Analytical Tools transmitted and received information relating to the software of the target machines, while the Virtual Reality Environment possessed functions for stereographic representation as well as reception of input from the user.

The benefits of visual analysis allow verification of the access structure and the hierarchical characteristics of software programs as well as the validation of software behavior, system performance evaluation, and detection of bugs, etc. Data for visual analysis was generated from the source codes and execution traces of target machine software. The Analytical Tools extracted the tasks and the modules from the source codes, analyzed the relations among them, organized this information in a hierarchical fashion, and prepared the static structural data. The

0-7803-1363-1/93 $3.00

Virtual Reality Environment then created visual representations from this data. With respect to visual design, the study focused on designing, editing, and correcting software for the target machines through visual representations in the Virtual Reality Environment.

3. 3-D Representations of Software

Three-dimensional graphical representations on stereographic displays were applied in developing a method for software visualization in the Virtual Reality Environment. The main advantage of using three-dimensional graphics was due to the reason that effective representations employing all three coordinate axes (X,Y, and Z) was possible.

Various software visualization methods were considered with a view to exploiting the advantages of three-dimensional graphics (Figure 2). Figure 2a) illustrates a method whereby the message flow among the tasks is represented in the X-Y plane, while the timing in which the tasks are executed is represented in the Z-axis. Thus, the results of inter-task static structural analysis can be represented as a block diagram in the X-Y plane while the results of behavioral

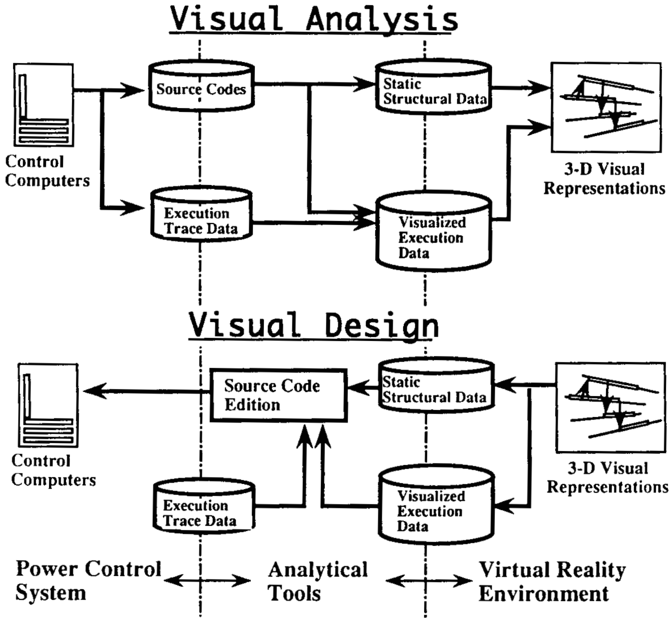

Figure 1. Visual Analysis & Visual Design

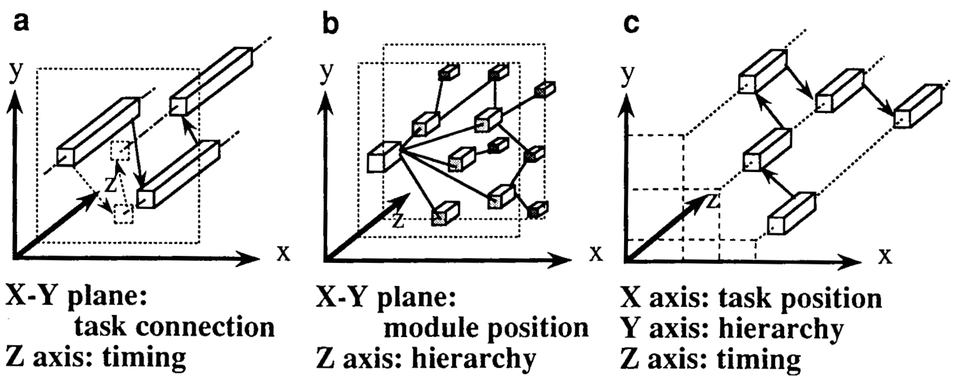

Figure 2. Methods of 3-D Representation

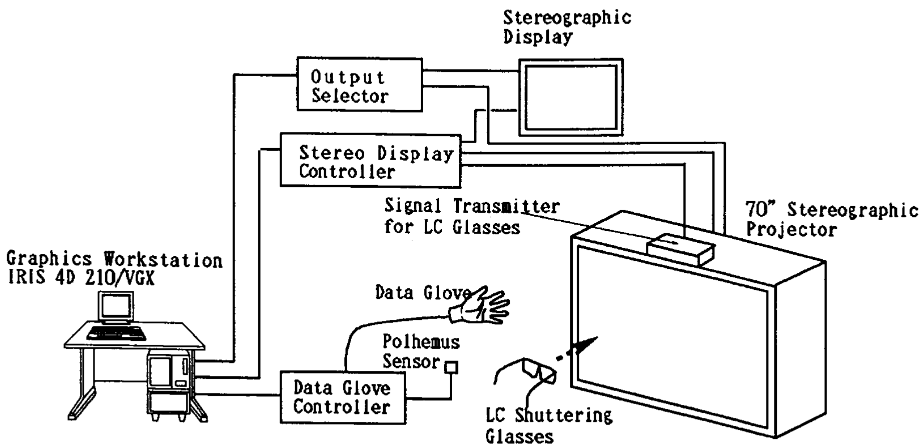

Figure 3. The Hardware Setup for the Virtual Reality Environment

analysis can be represented by a timing chart along the Z-axis. Figure 2b) illustrates a method for representing the hierarchical structure of the modules. In this case, the direction of the Z-axis represents the module hierarchy, while in the X-Y plane mapping is used in order to facilitate visual comprehension. In Figure 2c), a time chart has been added to Figure 2b); this method is used principally for the behavioral analysis of modules.

4. User Interfaces in the Virtual Reality Environment

The setup for the Virtual Reality Environment is shown in Figure 3. A stereographic projector was used to display respective graphical representations to the left and right eyes. A Data Glove was used as the three-dimensional input device. They constituted a user interface which allowed graphical representations to be handled in a manner similar to the manipulation of physical objects. Parts of graphical representations in the three-dimensional virtual world could be selected by pointing or moved by grasping as though they were objects in real world.

Using the Data Glove for input procedures permitted a high degree of flexibility compared to simultaneous operation with a mouse or a keyboard. This flexibility motivated the idea of utilizing hand gestures as operational commands. Eleven varieties of gestures were defined in the Virtual Reality Environment. They enabled the operator to perform almost all the necessary operations via the Data Glove, except for the input of alphabetical characters.

5. Utilization of 3-D Sound

The Virtual Reality Environment can offer more information to the user by utilizing not only visual, but also auditory media. The integration of three-dimensional sound to accompany stereographic display would permit more effective output to the user than those obtainable with graphics alone. Three-dimensional sound enables the user to perceive sounds as if they were actually emanating from various arbitrary spatial locations.

Hence, information not included within the scope of the screen, which cannot be displayed by the projector, can be transmitted using stereophonic sound; thereby permitting the generation of sounds from locations beyond the visual range. Three-dimensional sound can also be applied as a means for displaying a location of special interest within software visualization. It can be effectively utilized for complex representation not readily comprehended through graphics.

The development of techniques for three-dimensional acoustic representation of software is currently under way. As a primary application for the Virtual Reality Environment, a specific position in the virtual world could be displayed through sound. For example, when some patterns of graphical representations were searched for, a type of indicative sound would be generated as though it was emitted from the point of interest. Moreover, outlines of their shapes could be displayed by assigning an individual sound to each type of pattern.

6. Visual Analysis of Power Control Software

An example of the behavioral analysis of tasks in a power network control system is shown in Figure 4. In these images, the observation point is located along the X-axis direction such that time appears to advance from left to right. In order to indicate this, a time scale was drawn in the center of the display. In Figure 5, the view point can be easily changed within the image space so that the user can view the graphical representation as if he were located at a position inside the structure.

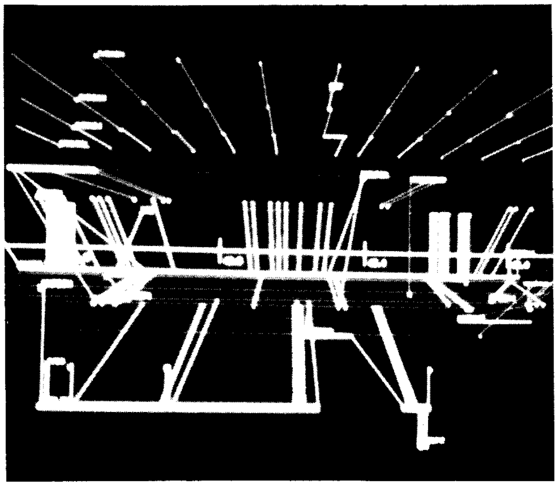

Figure 4. Visual Analysis of Power Control System Behavior

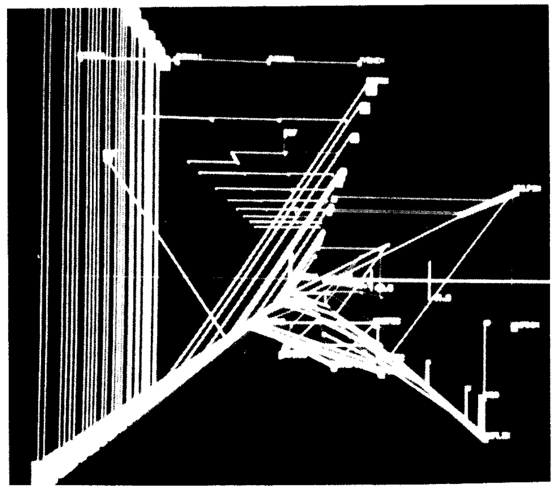

Figure 5. An Inner View for Visual Analysis

Figure 6 shows an example of static structural analysis. This is a stereographic representation of the access relations among all the modules constituting a task. In this diagram, the direction of the X-axis represents the hierarchical level of the access relations, and nesting grows deeper as one progresses from left to right. Thus, three-dimensional graphics can be effectively utilized by selecting the method of representation optimal for a particular purpose of analysis.

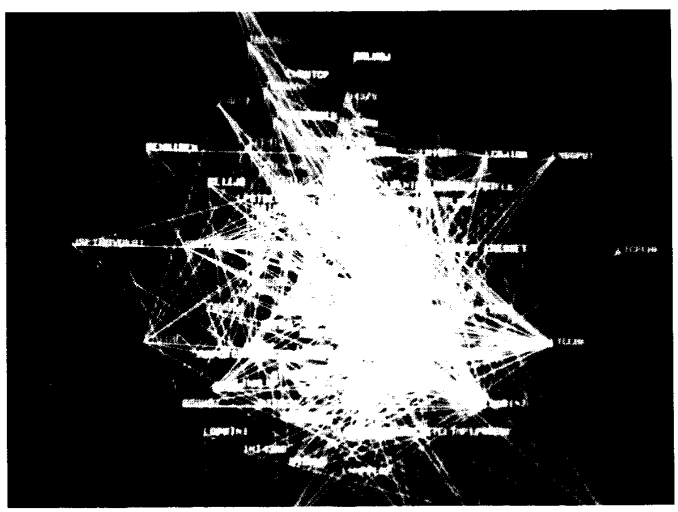

Figure 6. Static Structural Analysis of Modules

7. Application to Visual Design

The main purpose of Visual Design is to support the development of software; including software design, programming, testing, and debugging, etc. Each field is currently in the research stage. At present, this prototype Virtual Reality Environment allows the user to change graphical representations for the improvement of software structures and execution time.

Each function in a subsystem is realized as a sequence of tasks. If the timing of the execution of a first task is altered, the execution time of the subsequent tasks are also changed accordingly. Moreover, the execution time of other functions and their tasks will also be affected (Figure 7). As a result of using the Virtual Reality Environment, automatic estimation and visualization of changes in timing and prediction of the time required for overall execution of prescribed functions was enabled. The timing could be altered by grasping the portion representing the task with the Data Glove and shifting it along the time axis.

Even functions which proceed correctly in accordance with a schedule may no longer proceed correctly if the timing is changed because of a possible deadlock situation. Figure 8 shows a simplified version of an example where operation has terminated owing to a deadlock. In this figure, two processing sequences represented by X and Y involve the reading of data from files A and B, but these files are read in opposite order in sequences X and Y. In such a case, according to a certain timing, at the stage where processing sequence X has first accessed file A and is subsequently attempting to access file B, processing sequence Y is accessing file B. Hence, sequence X must wait until file B is released. However, processing sequence Y is also in a similar state, that is, awaiting the release of file A, and consequently neither X or Y can proceed. The prediction and avoidance of such faults can be studied through the Virtual Reality Environment.

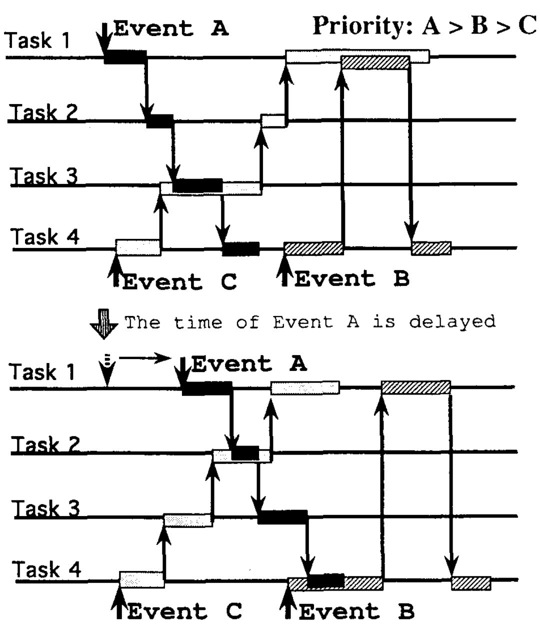

Figure 7. Estimation of Execution Time

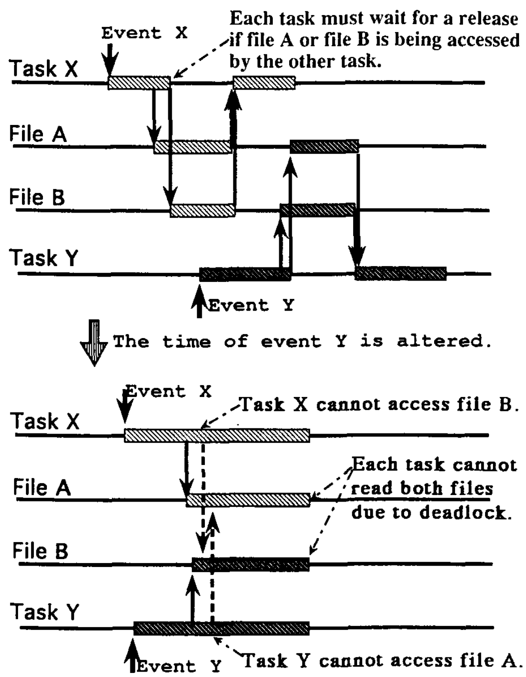

Figure 8. An Example of Deadlock

Accurate timing estimation necessitates overall assessment of the processing priority of each task, synchronization or exclusive control of processing operations, as well as CPU loads or communication loads, etc. At present, priorities, exclusive characteristics, and order of processing can be addressed, but the management of the other conditions is still in the research stage.

For other functions of Visual Design, any part of visual representations can be cut, copied and pasted by grasping and moving them with the Data Glove. At present, an investigation of the methods to reconstruct software structures or orders of execution by changing connection lines between boxes assigned to units of software is in progress.

8. Future Prospects

To build software for a large and complicated system such as a power distribution system that includes large numbers of computers, a greater quantity and variety of information should be displayed on the screen in a manner that is effective for users. Three-dimensional graphics can realize this through the assignment of different symbolic meanings to the three (X, Y,and Z) axes. Moreover, Virtual Reality techniques such as three-dimensional sound and spatial input devices also facilitate the recognition of the spatial position and orientation of graphical objects under consideration. The utilization of Virtual Reality technology permits effective comprehension for the operators by allowing them to enter the environment of this information and simulate manipulations such as moving and changing the shapes of physical objects[5]. The necessity of methods for providing the user with a large quantity of information will continually increase in the field of software development. Exploitation of three-dimensional visual and auditory display will inevitably address this growing need.

User-machine interfaces created by Virtual Reality technology have not yet attained a practical level with regards to cost, efficiency, and operating convenience. In particular, further improvement in the accuracy of the input devices is necessary. Moreover, improvement of the visualization methods and development of representations more readily comprehensible to the user still constitute problems for subsequent study.

Consequently, Virtual Reality technology can be effectively utilized as a means for supporting the development of large-scale software systems, and further enhancement of their effectiveness can be expected accordingly with the future development of hardware.

References

(1) Myoi, T., Amari, H., Inamura, K., Koike, H., Kuzuoka, H., Hirose, M., Ishii, T., & Hayashi, T.: "A Method of Large-Scale Control System Design Aided by System Visualization Technology", Electrical Engineering in Japan, Vol.112, No.4 (1992)

(2) Hirose, M., & Amari, H.: "A Study on Visualization of Control System", MIT-JSME Workshop on Cooperative Product Development, (1989)

(3) Hirose, M., Myoi, T., Amari, H., Inamura, K., & Stark, L.: "Development of Visual 3D Virtual Environment for Control Software", Human Machine Interfaces for Teleoperators & Virtual Environments, NASA Conference Publication 10071, (1990)

(4) Shu, N. C.: "Visual Programming", Van Nostrand Reinhold Company, New York (1988)

(5) Hirose, M., Myoi, T., Liu, A.,& Stark, L.: "Object Manipulation in Virtual Environment", Proc. of 6th Symposium on Human Interface, SICE (1990)